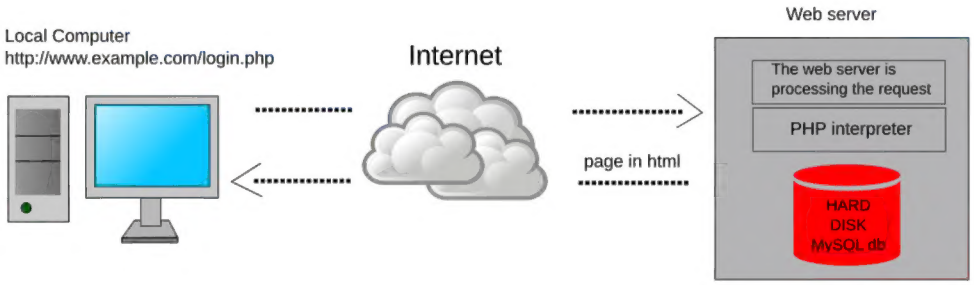
Dynamic web page

A **dynamic web page** is a web page constructed at runtime (during software execution), as opposed to a *static web page*, delivered as it is stored.



Dynamic web page: example of server-side scripting (PHP and MySQL).

A **server-side dynamic web page** is a web page whose construction is controlled by an application server processing server-side scripts.[1] In server-side scripting, parameters determine how the assembly of every new web page proceeds, and including the setting up of more client-side processing.

A **client-side dynamic web page** processes the web page using JavaScript running in the browser as it loads. JavaScript can interact with the page via Document Object Model (DOM), to query page state and modify it. Even though a web page can be dynamic on the client-side, it can still be hosted on a static hosting service such as GitHub Pages or Amazon S3 as long as there is not any server-side code included.

A dynamic web page is then reloaded by the user or by a computer program to change some variable content. The updating information could come from the server, or from changes made to that page's DOM. This may or may not truncate the browsing history or create a saved version to go back to, but a *dynamic web page update* using AJAX technologies will neither create a page to go back to, nor truncate the web browsing history forward of the displayed page. Using AJAX, the end user gets one dynamic page managed as a single page in the web browser while the actual web content rendered on that page can vary. The AJAX engine sits only on the browser requesting parts of its DOM, *the* DOM, for its client, from an application server. A particular application server could offer a standardized REST style interface to offer services to the web application.[2]

DHTML is the umbrella term for technologies and methods used to create web pages that are not static web pages, though it has fallen out of common use since the popularization of AJAX, a term which is now itself rarely used. Client-side-scripting, server-side scripting, or a combination of these make for the dynamic web experience in a browser.

Basic concepts

Classical hypertext navigation, with HTML or XHTML alone, provides "static" content, meaning that the user requests a web page and simply views the page and the information on that page.

However, a web page can also provide a "live", "dynamic", or "interactive" user experience. Content (text, images, form fields, etc.) on a web page can change, in response to different contexts or conditions.

There are two ways to create this kind of effect:

- Using client-side scripting to change interface behaviors *within* a specific web page, in response to mouse or keyboard actions or at specified timing events. In this case the dynamic behavior occurs within the presentation.
- Using server-side scripting to change the supplied page source *between* pages, adjusting the sequence or reload of the web pages or web content supplied to the browser. Server responses may be determined by such conditions as data in a posted HTML form, parameters in the URL, the type of browser being used, the passage of time, or a database or server state.[3]

Web pages that use client-side scripting must use presentation technology broadly called rich interfaced pages. Client-side scripting languages like JavaScript or ActionScript, used for Dynamic HTML (DHTML) and Flash technologies respectively, are frequently used to orchestrate media types (sound, animations, changing text, etc.) of the presentation. The scripting also allows use of remote scripting, a technique by which the DHTML page requests additional information from a server, using a hidden Frame, XMLHttpRequests, or a web service.

Web pages that use server-side scripting are often created with the help of server-side languages such as PHP, Perl, ASP, JSP, ColdFusion and other languages. These server-side languages typically use the Common Gateway Interface (CGI) to produce *dynamic web pages*. These kinds of pages can also use, on the client-side, the first kind (DHTML, etc.).

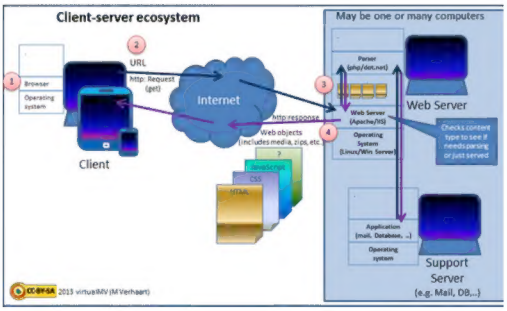
History

It is difficult to be precise about "dynamic web page beginnings" or chronology because the precise concept makes sense only after the "widespread development of web pages". HTTP has existed since 1989, HTML, publicly standardized since 1996. The web browser's rise in popularity started with Mosaic in 1993. Between 1995 and 1996, multiple dynamic web products were introduced to the market, including Coldfusion, WebObjects, PHP, and Active Server Pages.

The introduction of JavaScript (then known as LiveScript) enabled the production of client-side dynamic web pages, with JavaScript code executed in the client's browser.[4] The letter "J" in the term AJAX originally indicated the use of JavaScript, as well as XML. With the rise of server side JavaScript processing, for example, Node.js, originally developed in 2009, JavaScript is also used to dynamically create pages on the server that are sent fully formed to clients.

MediaWiki, the content management system that powers Wikipedia, is an example for an originally server-side dynamic web page, interacted with through form submissions and URL parameters. Throughout time, progressively enhancing extensions such as the visual editor have also added elements that are dynamic on the client side, while the original dynamic server-side elements such as the classic edit form remain available to be fallen back on (graceful degradation) in case of error or incompatibility.

Server-side scripting



When a request to a dynamic web page or resource is made, the application server processes the request using its server-side language.

A program running on a web server is used to generate the web content on various web pages, manage user sessions, and control workflow. Server responses may be determined by such conditions as data in a posted HTML form, parameters in the URL, the type of browser being used, the passage of time, or a database or server state.

Such web pages are often created with the help of server-side languages such as ASP, ColdFusion, Java, JavaScript, Perl, PHP, Ruby, Python, and other languages, by a **support server** that can run on the same hardware as the web server. These server-side languages often use the Common Gateway Interface (CGI) to produce dynamic web pages. Two notable exceptions are ASP.NET, and JSP, which reuse CGI concepts in their APIs but actually dispatch all web requests into a shared virtual machine.

The server-side languages are used to embed tags or markers within the source file of the web page on the web server.[5] When a user on a client computer requests that web page, the web server interprets these tags or markers to perform actions on the server. For example, the server may be instructed to insert information from a database or information such as the current date.

Dynamic web pages are often cached when there are few or no changes expected and the page is anticipated to receive considerable amount of web traffic that would wastefully strain the server and slow down page loading if it had to generate the pages on the fly for each request.

Client-side scripting

Client-side scripting is changing interface behaviors within a specific web page in response to input device actions, or at specified timing events. In this case, the dynamic behavior occurs within the presentation. The client-side content is generated on the user's local computer system.[6]

Such web pages use presentation technology called rich interfaced pages. Client-side scripting languages like JavaScript or ActionScript, used for Dynamic HTML (DHTML) and Flash technologies respectively, are frequently used to orchestrate media types (sound, animations, changing text, etc.) of the presentation. Client-side scripting also allows the use of remote scripting, a technique by which the DHTML page requests additional information from a server, using a hidden frame, XMLHttpRequests, or a Web service.

The first public use of JavaScript was in 1995, when the language was implemented in Netscape Navigator 2, standardized as ECMAScript two years later.[7]

Example

The client-side content is generated on the client's computer. The web browser retrieves a page from the server, then processes the code embedded in the page (typically written in JavaScript) and displays the retrieved page's content to the user.[8]

The `innerHTML` property (or write command) can illustrate the client-side dynamic page generation: two distinct pages, A and B, can be regenerated (by an "event response dynamic") as `document.innerHTML = A` and `document.innerHTML = B`; or "on load dynamic" by `document.write(A)` and `document.write(B)`.

Combination technologies

All of the client and server components that collectively build a dynamic web page are called a web application. Web applications manage user interactions, state, security, and performance.[9]

Ajax uses a combination of both client-side scripting and server-side requests. It is a web application development technique for dynamically interchanging content, and it sends requests to the server for data in order to do so. The server returns the requested data which is then processed by a client-side script. This technique can reduce server load time because the client does not request the entire webpage to be regenerated by the server's language parser; only the

content that will change is transmitted. Google Maps is an example of a web application that uses Ajax techniques.

A web client, such as a web browser, can act as its own server, accessing data from many different servers, such as Gopher, FTP, NNTP (Usenet) and HTTP, to build a page. HTTP supports uploading documents from the client back to the server. There are several HTTP methods for doing this.

See also

- Deep web (search indexing)
- Dynamic HTML
- htmx
- Personalization management system
- Responsive Web Design
- Solution stacks to serve dynamic web pages
  - LAMP (software bundle)
  - LYME (software bundle)
  - LYCE (software bundle)
- Static web page
- Web content management system
- Web template system

References

1. DeGroote, Sandy. "Subject and Course Guides: Digital UIC: Static and Dynamic Websites" (https://researchguides.uic.edu/digital-uic/static-dynamic) . *researchguides.uic.edu*. Retrieved 2023-12-12.
2. Fielding, Roy Thomas (2000). "Chapter 5: Representational State Transfer (REST)" (http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm) . *Architectural Styles and the Design of Network-based Software Architectures* (Ph.D.). University of California, Irvine.
3. Nelson, Anne; Nelson, William H. M. (2002). *Building Electronic Commerce with Web Database Constructions* (https://books.google.com/books?id=PU1PAAAAMAAJ) . Addison Wesley. ISBN 9780201741308.

4. Benson, Brent W. (1999). "JavaScript". *ACM SIGPLAN Notices*. **34** (4): 25–27. doi:10.1145/312009.312023 (https://doi.org/10.1145%2F312009.312023) . S2CID 10076062 (https://api.semanticscholar.org/CorpusID:10076062) ..
5. Tatroe, Kevin (2013). *Programming PHP: Creating Dynamic Web Pages* (3 ed.). O'Reilly. ISBN 9781449365837.
6. Doyle, Barry; Lopes, Cristina Videira (17 January 2008). "Survey of Technologies for Web Application Development". arXiv:0801.2618 (https://arxiv.org/abs/0801.2618) [cs.SE (https://arxiv.org/archive/cs.SE)].
7. Okin, J. R. (2005). *The Information Revolution: The Not-for-dummies Guide to the History, Technology, and Use of the World Wide Web* (https://archive.org/details/informationrevol0000okin) . Ironbound Press. ISBN 9780976385745.
8. "Learning VBScript", P. Lomax. ISBN 1-56592-247-6. Ed. O'Reilly, 1997. sec. C13.
9. "Web application" (https://web.archive.org/web/20140222134142/http://docforge.com/wiki/Web_application) . Archived from the original on 22 February 2014. Retrieved 24 July 2013.

External links

- Static versus dynamic web site (http://www.w3.org/wiki/How_does_the_Internet_work#Static_vs._Dynamic_Web_Sites) from W3.org
- Dynamic web sites (http://www.w3.org/People/Massimo/papers/rmm.pdf) using the Relationship Management Method, from W3.org.
- Wide analysis of dynamic web pages (https://www.ischool.utexas.edu/~hristova/ia/) from University of Texas, Austin.